TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Tietokonetekniikka

Tutkintotyö

Petri Mielonen

MURTOHÄLYTIN PIC16F870-MIKROKONTROLLERILLA

Työn valvoja Ilkka Tervaoja

Tampere 2007

TAMPEREEN AMMATTIKORKEAKOULU TUTKINTOTYÖ
Petri Mielonen

Mielonen, Petri Murtohälytin PIC16F870-mikrokontrollerilla
Tutkintotyö 22 sivua
Työn valvoja Lehtori Ilkka Tervaoja
Toukokuu 2007
Hakusanat mikrokontrolleri, 16F870, murtohälytin

TIIVISTELMÄ

Tämän tutkintotyön aiheena oli mikrokontrollerin avulla toteutettavan murtohälyttimen prototyypin suunnittelu. Suunnittelu painottui ohjelman kehittämiseen ja erilaisten ideoiden kokeilemiseen, joten laitteesta tehtiin melko yksinkertainen ja vain tärkeimpiä toimintoja ja niiden kokeilua palveleva testausalusta. Laitteen suunnittelusta jätettiin pois muutamia sellaisia asioita, jotka olisivat valmiin laitteen varsinaisessa käytössä tarpeellisia, mutta eivät prototyypin kehittämisen kannalta välttämättömiä.

Työn tuloksena saatiin laite, joka toiminnoiltaan ja ominaisuuksiltaan vastaa hyvin asetettuja tavoitteita. Työn aikana tuli myös esiin joitakin asioita, joissa olisi tarvetta jatkokehitykseen. Syntyi myös useita ideoita laitteen ja sen toimintojen viemisestä vielä monipuolisemmaksi.

TAMPERE POLYTECHNIC THESIS
Computer systems engineering
Embedded systems

Mielonen, Petri PIC16F870 based burglar alarm
Engineering Thesis 22 pages
Thesis Supervisor Lecturer Ilkka Tervaoja
May 2007
Keywords microconrtoller, 16F870, burglar alarm

ABSTRACT

The subject of this thesis was to design a prototype of a microcontroller-based burglar alarm. The primary focus was on the software, while the hardware was made mainly in order to test the software. The hardware was intentionally kept quite simple. Some features, which are not so important for testing purposes, but would be necessary in actual use of the system, were left out of the subject.

The result of this study was a working alarm system prototype, that meets the goals set. However, some ideas to make the alarm system more functional were found. Also there were some issues found that would need futher improvement.

ALKUSANAT

Tutkintotyön aihe pohjautui kiinnostukseen ja aiempaan kokemukseen mikrokontrollereiden käytöstä. Myös kiinnostus turva- ja hälytinjärjestelmiä kohtaan ja käytettävissä ollut riittävä välineistö, mukaan luettuna sopiva mikrokontrollerin ohjelmointilaite, tukivat aiheeseen päätymistä. Työn tein pääasiassa kotona, mutta myös Tampereen ammattikorkeakoulun tietokonetekniikan laboratoriotiloissa suoritin joitakin mittauksia. Työtä suunnittelin alustavasti jo syksyllä 2006, mutta varsinainen työskentely tapahtui kevään 2007 kuluessa.

Haluan kiittää Tampereen ammattikorkeakoulun opettajia hyvästä ja asiantuntevasta opetuksesta ja koulun muuta henkilökuntaa laadukkaan opiskeluympäristön ylläpidosta. Erityisesti kiitän tämän työn valvojaa, lehtori Ilkka Tervaojaa ohjauksesta ja työn tarkastamisesta.

Tampereella 28. toukokuuta 2007

_____

Petri Mielonen

SISÄLLYSLUETTELO

KÄYTETYT LYHENTEET

PIC	Peripheral interface controller
DIP	Dual in-line package
SOIC	Small-outline integrated circuit
SSOP	Shrink small-outline package
PLCC	Plastic leaded chip carrier
TQFP	Thin quad flat pack
USART	Universal synchronous asynchronous receiver transmitter
PWM	Pulse width modulation
I/O	Input/output
ppm	Parts per million

1 JOHDANTO

Työn tavoitteena oli suunnitella ja toteuttaa mikrokontrolleriin perustuva murtohälyttimen prototyyppi ja sen ohjelma. Työn laajuus rajattiin hälyttimen ydintoimintoihin ja erityisesti sen ohjelman kehittämiseen, joten useita laitteen todellisen käytön edellyttämiä osia jätettiin työn ulkopuolelle. Näitä ovat esimerkiksi sähkönsyöttö akkuvarmennuksineen ja laitteen kotelointi.

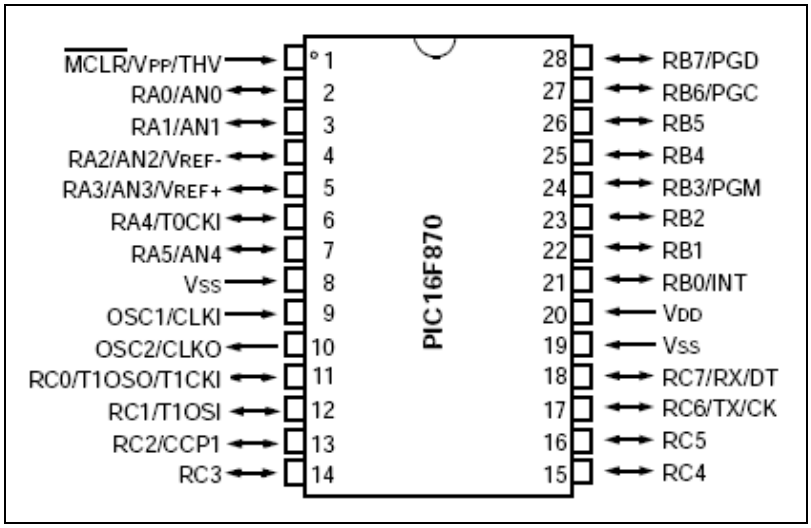
Laitteen toimintojen suunnittelussa on jonkin verran huomioitu yleisesti käytössä olevien kaupallisten tuotteiden tyypillisiä toimintaperiaatteita, kuten päätevastusten käyttöä valvontasilmukoissa, mutta yksityiskohdiltaan laite on täysin tekijän omiin ideoihin ja kokeiluihin perustuva. Ohjelman suunnittelussa ei ole käytetty perustana mitään valmiita ratkaisuja.

Suunniteltuun laitteeseen on valittu 16F870-mikrokontrolleri I/O-nastojen riittävän määrän, A/D-muuntimen, hyvän saatavuuden ja käytettävissä olleen ohjelmointilaitteen perusteella. Ohjelma on kirjoitettu assembly-kielellä käyttäen Microchipin MPLAB-kehitysympäristöä.

Aluksi esitetään laitteen toimintojen kannalta keskeisimmät mikrokontrollerin ominaisuudet ja toiminnot. Tämän jälkeen selvitetään laitteen rakenne ja sen toiminta.

2 PIC16F870

PIC16F870 on Microchipin valmistama kahdeksanbittinen FLASH-tekniikkaan perustuva mikrokontrolleri. Piiri on 28-nastainen ja sitä on saatavana DIP-, SOIC-, SSOP-, PLCC- ja TQFP-koteloisena. Kontrollerissa on kaksi kilotavua FLASH-ohjelmamuistia, 128 tavua datamuistia ja 64 tavua EEPROM-muistia. Kontrolleri toimii 0...20 MHz:n kellotaajuudella. Piirissä on kolme porttia (portit A, B ja C), joissa on yhteensä 22 I/O-liitäntää. /1, s. 1-14/



Kuva 1. 16F870-kontrollerin nastajärjestys /1, s. 2/

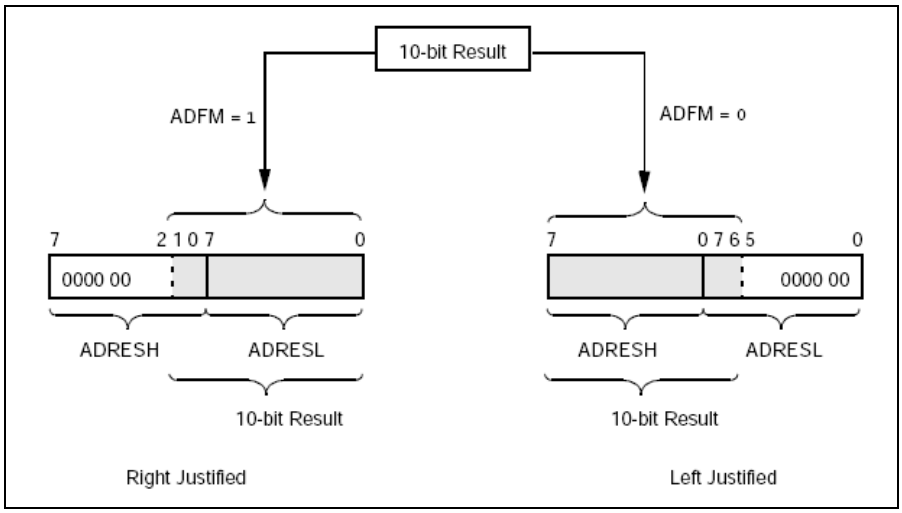
2.1 I/O-liitännät

Kontrollerin jokainen I/O-nasta on valittavissa sisään- tai ulostuloksi. Valinta tehdään asettamalla rekisterien TRISA, TRISB ja TRISC bitit ykköseksi tai nollaksi. Ulostuloksi asetettujen nastojen alkutila määrätään PORTA-, PORTB- ja PORTC-rekisterien avulla. Joissakin nastoissa on myös vaihtoehtoisia toimintoja, kuten USART, PWM tai Timer1-oskillaattoriulostulo. Porttia A voidaan käyttää myös A/D-muuntimena, jossa on enimmillään viisi kanavaa. Portissa B on sisäiset ylösvetovastukset, jotka voidaan ottaa ohjelmallisesti käyttöön. /1, s. 1-14/

2.2 A/D-muunnin

Portti A voidaan konfiguroida A/D-muuntimeksi. Rekisterin ADCON1 avulla määrätään, mitkä nastat ovat analogiatuloja ja mitä nastoja käytetään muuntimen referenssijännitteiden asettamiseen. Analogiatuloiksi määriteltävien nastojen tulee olla asetettuina sisääntuloiksi, muuten muunnostulos on ulostulonastan tila. Referenssijännitteiksi voidaan myös valita piirin syöttöjännite ja maapotentiaali, jolloin saadaan vielä useampi nasta A/D-kanavaksi. A/D-muunnokseen kulloinkin käytettävä kanava valitaan rekisterillä ADCON0. Saman rekisterin biteillä valitaan myös muunnoksen nopeus (vaikuttaa tarkkuuteen), aloitetaan muunnos ja käynnistetään tai sammutetaan koko A/D-muunninmoduuli. /1, s. 79-81/

Muunnoksen valmistuminen voidaan havaita joko keskeytyksen avulla tai tarkkailemalla ADCON0-rekisterin bittiä 2, joka nollautuu, kun muunnos on valmis. Muunnostulos on 10-bittinen, ja se kirjoitetaan ADRESH- ja ADRESL-rekistereihin, jotka ovat kahdeksanbittisiä. ADCON1-rekisterin bitillä 7 valitaan, tasataanko tulos ADRESH-rekisterin eniten merkitsevään vai ADRESHL-rekisterin vähiten merkitsevään bittiin (kuva 2). Käyttämättä jäävät kuusi bittiä kirjoitetaan nolliksi. /1, s. 79-81/



Kuva 2. A/D-muunnostuloksen tasaaminen /2/

2.3 Keskeytykset

Keskeytyksiä hallitaan INTCON-rekisterin (interrupt control) avulla. Se sisältää useita keskeytysten sallintabittejä ja lippubittejä, jotka asettuvat tapahtuneen keskeytyksen merkiksi. INTCON-rekisterin bitti 7 on GIE (global interrupt enable), jolla sallitaan tai kielletään kaikki keskeytykset. Myös rekistereissä PIE1 ja PIE2 on keskeytysten sallintabittejä, ja niitä vastaavien keskeytysten lippubitit ovat rekistereissä PIR1 ja PIR2. /1, s. 18-22, 96-98/

Keskeytys voidaan aiheuttaa usealla eri tapahtumalla, esimerkiksi Timer0:n tai Timer1:n ylivuodolla, portti B:n nastojen tilanmuutoksilla tai A/D-muunnoksen valmistumisella. Kun keskeytys tapahtuu, GIE-bitti nollautuu estämään muut keskeytykset, ohjelman paluuosoite pannaan pinoon ja keskeytysrutiinin alkuosoite 0004h ladataan ohjelmalaskuriin. Useita eri keskeytyksiä käytettäessä keskeytyksen aiheuttanut tapahtuma selvitetään lippubittien avulla. Lippubitit on nollattava ennen keskeytyksestä poistumista, jotta ne eivät aiheuta välittömästi uutta keskeytystä. Keskeytysrutiinista poistutaan käskyllä retfie, joka sallii keskeytykset asettamalla GIE-bitin ja palaa jatkamaan ohjelman suoritusta pinoon pannusta osoitteesta. /1, s. 18-22, 96-98/

2.4 STATUS-rekisteri

STATUS-rekisteri sisältää zero- sekä carry- ja borrow-lippubitit, joita käytetään matemaattisten operaatioiden yhteydessä. Zero-lipun avulla voidaan päätellä, oliko suoritetun laskutoimituksen tulos nolla. Carry- ja borrow-bitillä havaitaan yhteenlaskusta aiheutunut ylivuoto tai laskutoimituksen negatiivinen tulos. Näitä lippuja käytetään ehdollisten toimintojen yhteydessä, esimerkiksi vähennettäessä jonkin rekisterin arvoa voidaan hyppykäsky tai aliohjelmakutsu suorittaa, mikäli rekisterin arvoksi tuli nolla. /1, s. 16/

2.5 W-rekisteri

PIC-mikrokontrollerissa käytetään työrekisteriä W (work register), jonka avulla lukuja käsitellään. Johonkin yleiskäyttöiseen rekisteriin lukua vietäessä ladataan se ensin W-rekisteriin, josta se kopioidaan haluttuun rekisteriin. Työrekisterin avulla suoritetaan myös matemaattisia operaatioita. Esimerkiksi käsky "addwf luku,0" laskee yhteen W-rekisterin ja luku-nimisen rekisterin. Operandilla 0 tulos kirjoitetaan W-rekisteriin. Jos operandi on 1, tulos ladataan luku-rekisteriin.

2.6 Timer1

Timer1 on kahdesta kahdeksanbittisestä rekisteristä (TMR1H ja TMR1L) koostuva ajastin ja laskuri, joka voidaan konfiguroida toimimaan ulkoisen kellopulssin tai kontrollerin sisäisen kellon tahdissa. Mikäli käytetään sisäistä kelloa, on sen toimintataajuutena kontrollerin kellotaajuus jaettuna neljällä. Lisäksi Timer1-moduulissa on esijakaja, jolla moduulin kellotaajuus voidaan jakaa kahdella, neljällä tai kahdeksalla. Timer1 konfiguroidaan ja kytketään toimintaan T1CON-rekisterin biteillä. /1, s. 49-52/

Timer1:n toimiessa ajastimena kasvaa sen arvo esijakajan mukaan jokaisella, joka toisella tai joka neljännellä ohjelman käskysuorituksella. Laskurina käytettäessä sen arvo kasvaa ulkoisen kellotulon nousevalla reunalla. Rekistereihin TMR1H ja TMR1L voidaan myös kirjoittaa. Timer1 laskee 0000h...FFFFh, minkä jälkeen laskeminen alkaa uudelleen nollasta. Tällöin aiheutuu Timer1 ylivuoto -keskeytys, mikäli se on sallittu PIE1-rekisterin bitillä 0. /1, s. 49-52/

3 MURTOHÄLYTIN

Työssä suunniteltiin ja rakennettiin murtohälyttimen prototyyppi, joka toiminnoiltaan ja ominaisuuksiltaan on samankaltainen kuin kaupalliset tuotteet.

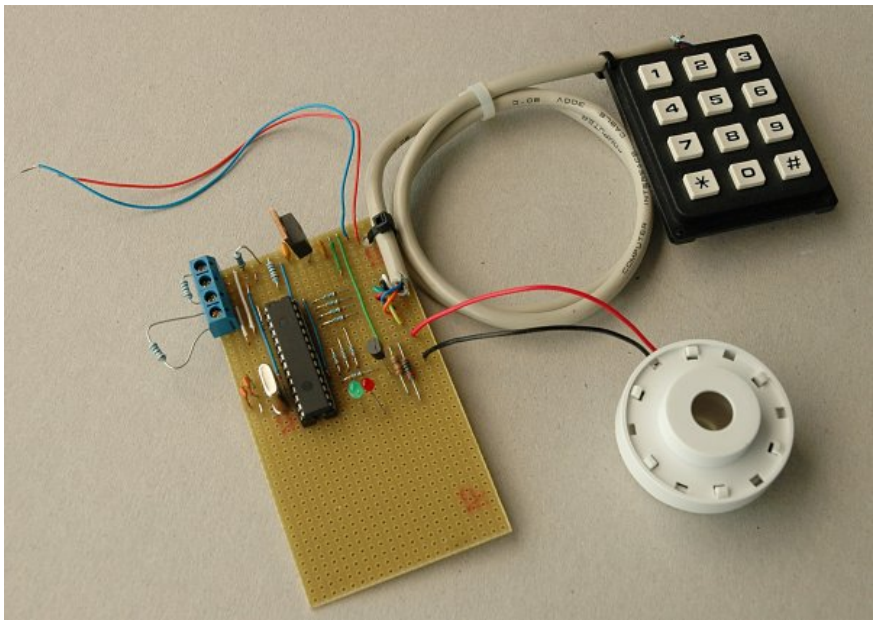
Hälyttimessä on näppäimistö, jolla syötetään koodinumero laitteen kytkemiseksi päälle tai pois päältä. Laitteeseen voidaan kytkeä kaksi päätevastuksilla varustettua valvontasilmukkaa, joihin voidaan liittää useita avautuvia tai sulkeutuvia kytkimiä. Silmukoista toinen on viiveellinen, mikä tarkoittaa, että havaitessaan muutoksen valvontasilmukassa se ei aiheuta välitöntä hälytystä, vaan aloitetaan tietyn mittaisen poiskytkentäviiveen laskenta. Viiveellistä silmukkaa käytetään valvomaan sitä tilaa, jossa joudutaan liikkumaan ennen hälyttimen pois kytkemistä ja sen kytkemisen jälkeen. Mikäli hälytintä ei valvottuun tilaan saapumisen jälkeen oikealla koodilla kytketä toiminnasta ohjelmassa määrätyn ajan kuluessa, aiheutuu hälytys. Toinen silmukka on viiveetön ja aiheuttaa hälytyksen heti. Näin ollen viiveetön silmukka voidaan asentaa vain sellaisten tilojen valvontaan, joiden kautta ei jouduta hälytyslaitteelle kulkemaan. Silmukoihin voidaan liittää erilaisia kytkimen kaltaisesti toimivia antureita ja tunnistimia, esimerkiksi magneettikytkimiä tai relelähdöllä varustettuja liikeilmaisimia.

3.1 Laitteen kytkentä

Laite rakennettiin aluksi koekytkentäalustalle, jolla sen ensimmäisiä versioita voitiin kokeilla ja kytkentää helposti muuttaa. Myöhemmin laite koottiin VERO-levylle (kuva 3). Laite koostuu näppäimistöstä, hälytyssireeninä toimivasta pietsosummerista, valvontasilmukoiden johtimista kytkimineen ja piirilevyllä olevasta kytkennästä. Kytkentä (kuva 4) on melko yksinkertainen ja sisältää vain tärkeimmät ohjelman kokeiluun ja kehittämiseen vaadittavat komponentit.

3.2 Näppäimistö

Näppäimistönä laitteessa on 12-painikkeinen matriisinäppäimistö. Se on liitetty kontrollerin I/O-porttiin B. Näppäimistö koostuu kolmesta sarakkeesta ja neljästä rivistä. Näppäimen painaminen yhdistää yhden rivin yhteen sarakkeeseen, joka voidaan havaita kontrollerin avulla. Portin nastat 1-3 on kytketty sarakkeisiin ja



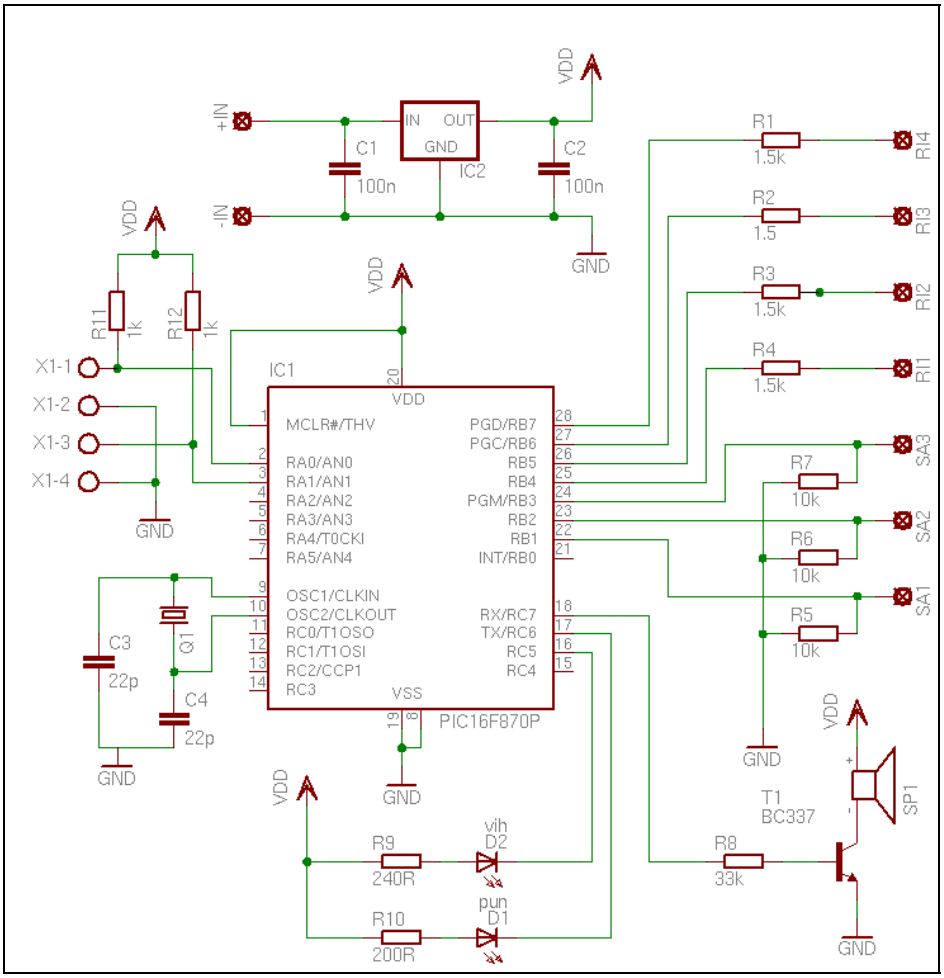
Kuva 3. VERO-levylle koottu laite

konfiguroitu sisääntuloiksi. Alasvetovastuksilla R5-R7 pidetään sarakejohtimien jännitteet nollassa silloin, kun mitään näppäintä ei ole painettu. Nastat 4-7 ovat ulostuloja ja ne on kytketty näppäimistön riveille. Vastukset R1-R4 estävät eri tiloissa olevien lähtönastojen välisen oikosulun, joka olisi ilman vastuksia mahdollinen silloin, kun painetaan vähintään kahta näppäintä samassa sarakkeessa samanaikaisesti.

Näppäimistön lukeminen tapahtuu asettamalla yksi portin B ulostulonasta (4-7) kerrallaan ylätilaan. Sarakkeita vastaavia sisääntuloja tarkkaillaan ohjelmallisesti, ja mikäli jokin näppäin on painettuna, näkyy painetun näppäimen saraketta vastaavassa sisääntulossa noin viiden voltin jännite. Kun kaikki sarakkeet on tarkistettu, lasketaan rivin jännite nollaksi ja nostetaan seuraavan rivin jännite ylätilaan. Näin jatketaan, kunnes kaikki rivit on tarkastettu.

3.3 Valvontasilmukat

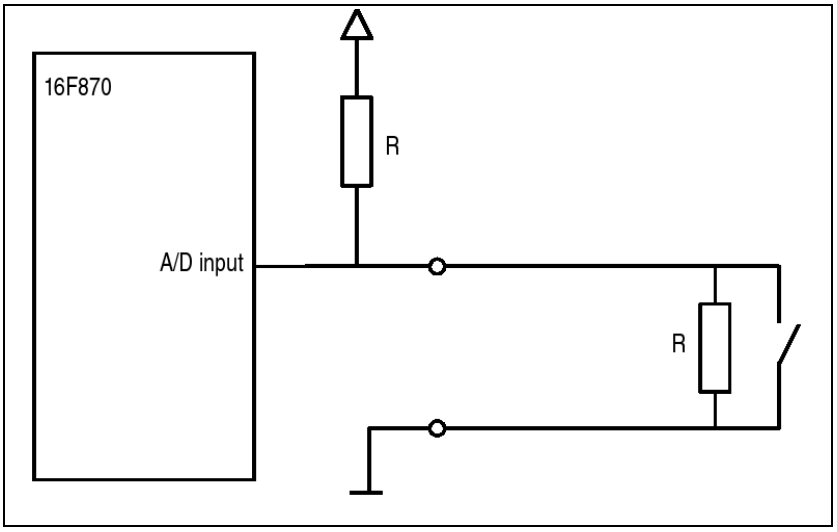
Silmukat on kytketty kontrollerin I/O-nastoihin RA0 ja RA1, jotka on konfiguroitu analogiatuloiksi. Kummassakin silmukassa on 1 kΩ:n päätevastus, jolla silmukan resistanssi asetetaan oikeaksi. Päätevastus on kytketty sarjaan toisen vastuksen



Kuva 4. Kytkentä

kanssa (kuva 5) ja näin muodostettu jännitejako synnyttää silmukkaan tietyn jännitteen, jonka suuruutta valvotaan kontrollerin analogiatulon avulla. Silmukan johtimien oikosulkeminen tai katkaiseminen muuttaa jännitettä merkittävästi. Valvottavat silmukat kytketään liittimeen X1. Viiveellinen silmukka kytketään napoihin 1 ja 2, viiveetön napoihin 3 ja 4. Silmukkaan voidaan kytkeä useita kytkimiä. Avautuvat kytkimet kytketään sarjaan päätevastuksen kanssa, sulkeutuvat sen rinnalle. Päätevastus on kytkettävä viimeiseen kytkimeen, muuten vastuksen jälkeen olevan piirin katkeamista ei havaita. Jos silmukkaa ei käytetä lainkaan, kytketään liittimeen ainoastaan päätevastus.

Silmukan johtimesta aiheutuvan resistanssin, käytettyjen vastusten toleranssien ja lämpötilan muutoksesta johtuvan vastusten resistanssien vaihtelun vuoksi täytyy silmukan jännitteellä olla riittävä, sallittu vaihteluväli. Hälytyksen aiheuttavat



Kuva 5. Silmukkavalvonnan periaatekuva

jännitteiden raja-arvot määritellään ohjelmassa. Silmukkavalvonnan jännitejaossa käytettävien 1 kΩ:n vastusten lämpötilakerroin on 100 ppm/°C, joten esimerkiksi 50 asteen lämpötilaero merkitsee resistanssissa 5000 miljoonasosan eli 0,005 kertaa vastuksen nimellisarvon eroa. Siten resistanssien ero on viisi ohmia.

3.4 Hälytyssummeri

Hälytysäänen synnyttävänä sireeninä käytetään pietsosummeria, joka on riittävän kovaääninen hälytystarkoitukseen. Summerin nimellinen käyttöjännite on 12 V, mutta se toimii hyvin pienemmälläkin jännitteellä. Virtaa summeri ottaa viiden voltin jännitteellä noin 10 mA. Summerin voisi kytkeä suoraan mikrokontrollerin lähtönastaan, jonka suurin sallittu virta kumpaan tahansa suuntaan on 25 mA /1, s. 117/. Summeria kuitenkin ohjataan kytkimenä toimivalla transistorilla, jotta tarvittaessa pietsosummerin tilalle voidaan helposti vaihtaa jokin muu hälytyslaite tai rele. Transistori (NPN) on tyypiltään BC337, ja sen suurin sallittu kollektorivirta on 800 mA /2/.

3.5 Muut komponentit

Oskillaattorina käytetään 4 MHz:n kideoskillaattoria. Sen kummankin nastan ja maapotentiaalin väliin on kytketty 22 nF:n kondensaattori varmistamaan, että oskillaattori alkaa värähdellä.

Punainen ja vihreä led etuvastuksineen antavat laitteen käyttäjälle tietoa laitteen tilasta. Punaisella ledillä ilmaistaan hälyttimen olevan aktiivisena tai poistumisviiveen olevan käynnissä. Vihreä led osoittaa laitteen olevan valmiustilassa.

Regulaattori on kytkennässä varmistamassa, että kytkennän syöttöjännite ei säädettävää jännitelähdettä käytettäessä vahingossa nouse liian suureksi.

4 HÄLYTTIMEN OHJELMA

Ohjelma on kirjoitettu assembly-kielellä Microchipin MPLAB-ohjelmistoa käyttäen. Sen avulla ohjelmaa on voitu myös simuloida, mikä on helpottanut ja nopeuttanut toimintojen kokeilemista ja ohjelman kehittämistä. Ohjelman aikasidonnaiset toiminnot perustuvat keskeytyksen käyttöön ja lippubittien asetteluun sekä niiden mukaan toimimiseen. Suurimman osan ajasta ohjelman suoritus on vain näppäimistön lukemista ja hälyttimen ollessa aktiivisena A/D-muunnosten tulkintaa.

4.1 Keskeytys ja ajastukset

Keskeytykseen käytetään timer1-ajastinta, joka aiheuttaa keskeytyksen ylivuototilanteessa. Timer1 saa kellopulssin kontrollerin sisäisestä kellosta. Timer1:n esijakaja on asetettu arvoon 1:1, joten timer1:n arvo suurenee oskillaattoritaajuuden neljäsosan eli 1 MHz:n taajuudella. Siten timer1:n arvo

suurenee yhdellä mikrosekunnin välein.

Keskeytysohjelman alussa kopioidaan STATUS- ja W-rekisterin sisältö apurekistereihin, joista ne voidaan ennen keskeytysrutiinista poistumista palauttaa. Näin on tehtävä, koska kyseisiä rekistereitä tarvitaan myös keskeytyksen aikana, eikä keskeytys saa missään tapauksessa sotkea rekistereissä keskeytyksen alkamishetkellä olevaa sisältöä.

Keskeytysrutiinissa lisätään rekisteriin TMR1H luku 00111100 ja rekisteriin TMR1L 10110001, jolloin timer1:n arvo on luvun 15537 (lisätty luku 16-bittisenä) ja ennen lisäystä rekisterien sisältämän luvun summa. Tämä johtaa uuteen keskeytykseen 50 millisekunnin kuluttua. Timer1:n arvo suurenee myös keskeytysrutiinin aikana, joten keskeytyksen suorittaminen ei sotke ajastusta. Keskeytysohjelma kestää vain noin 20-30 mikrosekuntia, joten ei ole vaaraa, että seuraava keskeytys viivästyisi edellisen ollessa vielä kesken.
Keskeytysohjelmassa asetetaan merkkilippu, jonka perusteella pääohjelmassa keskeytysohjelman suorituksen jälkeen tehdään 50 millisekunnin välein toistuvat rutiinit. Lippubitti nollataan, kun sen osoittamia tehtäviä aletaan suorittaa. Jokaisella keskeytyskerralla vähennetään TIMECOUNT-nimistä rekisteriä, jonka arvo on aluksi 10. Vähentämisen jälkeen suoritetaan aliohjelman "half_sec" kutsu, mikäli STATUS-rekisterin zero-lippu asettui, eli vähennyslaskun tulos oli nolla. Käytännössä tämä tarkoittaa sitä, että on kulunut puoli sekuntia siitä, kun TIMECOUNT-rekisterin arvoksi edellisen kerran ladattiin luku 10. Aliohjelmassa "half_sec" asetetaan TIMECOUNT-rekisterin arvoksi jälleen 10. Tähän aliohjelmaan perustuvat puolen sekunnin aikaa vaativat toiminnot, kuten punaisen ledin vilkkuminen poistumisviiveen aikana ja hälytyssummerin pulssimainen soiminen. Aliohjelman suorituskertoja lasketaan, ja joka toisella kerralla asetetaan sekunnin kulumista osoittava lippubitti. Sen perusteella pääohjelmassa kutsutaan sec-aliohjelmaa, jossa suoritetaan sekunnin kulumiseen sidottuja toimintoja, kuten useiden eri viiveaikojen laskentoja.

4.2 Näppäimistön lukeminen

Näppäimistö luetaan 50 millisekunnin välein skannausperiaatteella. Luenta aloitetaan keskeytysohjelmassa asetettavan lippubitin perusteella. Näppäimistön lukurutiinin alussa ladataan W-rekisteriin luku FFh. Tämän jälkeen näppäimistö käydään läpi rivi kerrallaan tutkimalla jokaisen rivin sarakkeet (esimerkki 1). Mikäli kyseisen rivin jollain sarakkeella on alas painettu näppäin, viedään kyseisen näppäimen numeroa vastaava luku W-rekisteriin. Viimeisen rivin jälkeen w-rekisterin luku ladataan NEWKEY-nimiseen rekisteriin. Jos mitään näppäintä ei luettu, on ladattava luku rutiinin alussa työrekisteriin ladattu FFh. Seuraavaksi suoritetaan kaksi ehtoa, joiden perusteella suoritetaan hyppykäsky, mikäli mitään näppäintä ei oltu painettu tai painettuna oli sama näppäin kuin edellisellä lukukerralla.

Uuden näppäilyn havaitsemisen jälkeen ladataan SECONDS-nimiseen rekisteriin luku 10. Tämän rekisterin arvoa vähennetään sekunnin välein (sec-aliohjelma, ks. kohta 4.1), ja rekisterin arvon tullessa nollaksi tyhjennetään jo painellut näppäilyt muistirekistereistä. Siten huolehditaan, että vahinkonäppäilyt eivät sotke myöhempiä koodin syöttöyrityksiä. Koodin vastaanottamisen lopettaa #-näppäin, jonka havaitseminen johtaa koodin tarkistukseen. Oikea koodi on EEPROM:ssa, josta se luetaan numero kerrallaan ja luettua numeroa verrataan näppäiltäessä tallennettuihin numeroihin. Mikäli oikea koodissa on vähemmän kuin kahdeksan numeroa, on EEPROM:ssa käyttämättömien numeroiden paikalla FEh.

Luettujen näppäilyjen tallettamiseen käytetään kahdeksaa rekisteriä, joihin osoitetaan FSR-osoitinrekisteriä (file select register) käyttäen. FSR-rekisteri osoittaa aluksi rekisteriin 0x38, ja jokaisen näppäimistöltä luetun numeron jälkeen osoitinta kasvatetaan incf-käskyllä. Kun koodi on tarkistettu tai kymmenen sekunnin näppäilyaika on kulunut loppuun, palautetaan osoitin alkutilaan ja kaikkiin numeroiden muistirekistereihin ladataan luku FEh.

4.3 A/D-muunnokset

Portin A nastat 2 ja 3 on konfiguroitu analogiatuloiksi, ja referenssijännitteinä käytetään kontrollerin syöttöjännitettä sekä nollapotentiaalia. Analogiatulojen jännitealue on siten 0-5 volttia, ja koska muunnostulos on 10-bittinen, jakautuu jännitealue 2^{10} eli 1024 tasoon. Muunnostulos tasataan ADRESH-rekisterin eniten merkitsevään bittiin, joten tuloksen kaksi vähiten merkitsevää bittiä on ADRESL-rekisterissä. A/D-muunnostuloksen tulkinnan helpottamiseksi tulosta käsitellään vain kahdeksanbittisenä, joten kaksi alinta bittiä jäävät merkityksettömiksi. Tästä aiheutuva virhe on enimmillään noin 20 mV, joka ei ole merkittävä.

A/D-muunnos suoritetaan silmukoille vuorotellen, ja ne tehdään keskeytysohjelman avulla 50 millisekunnin aikavälein. Tällä tavoin saadaan taattua riittävä aika A/D-muuntimen kondensaattorin varautumiseen sekä muunnoksen valmistumiseen. Vuorottelu sekä keskeytysohjelmaan tahdistaminen johtavat siihen, että kummankin kanavan muunnos tapahtuu 100 millisekunnin välein. Kun valmiin muunnoksen tulos on tarkasteltu, asetetaan muunnin mittaamaan toista kanavaa ja käynnistetään A/D-muunnos. Seuraavan keskeytyksen jälkeen luetaan edellisen muunnoksen tulos ja käynnistetään jälleen toisen kanavan mittaus.

Silmukan sallittu jännitealue on asetettu ohjelmassa noin 1,5-3,5 volttiin. Jännite saa siis vaihdella (johtimien resistanssi, lämpötilan vaikutukset ym.) kaksi volttia. Silmukan rajajännitteiden ylittyminen selvitetään vähentämällä ADRESH-rekisteristä (A/D-muunnoksen tulos) rajajännitettä vastaava luku ja päättelemällä STATUS-rekisterin carry- ja borrow-bitin avulla, oliko silmukan jännite suurempi vai pienempi kuin asetettu raja-arvo. Alarajajännitettä vastaava 8-bittinen binääriluku on 0100110011 ja ylärajajännitettä vastaava 1011001100. Binääriluku ladataan ensin W-rekisteriin, ja sen jälkeen suoritetaan käsky "subwf ADRESH,0", joka vähentää W-rekisterin sisällön rekisteristä ADRESH. Operandilla 0 tulos viedään W-rekisteriin. Kumpikin raja tarkastellaan siis vähentämällä asetettu raja-arvo mitatusta silmukan jännitteestä. Silmukan jännite on vähintään 1,5 V (alaraja), mikäli vähennyslaskun tulos on positiivinen ja alle 3,5 V (yläraja), jos tulos on

negatiivinen. Ylä- ja alaraja tarkastellaan peräkkäin, ja rajan ylittyessä asetetaan lippubitti, jonka perusteella mittauksen kohteena olevasta silmukasta riippuen joko käynnistetään hälytys heti tai aloitetaan viiveajan laskenta. Mikäli viiveellisen silmukan jännite on sallitun alueen ulkopuolella silmukan valvonta lopetetaan. Tällöin mitataan ainoastaan viiveettömän silmukan jännitettä, ja mikäli siinä havaitaan kielletty jännite, tehdään hälytys heti.

4.4 Viiveajat

Ohjelmassa on neljä rekisteriä viiveaikojen laskentaa varten. Niiden avulla lasketaan hälyttimen poiskytkentäviive, poistumisviive, näppäilyaika ja pisin hälytysaika.

Viiveiksi on kokeilukäyttöä varten asetettu kymmenen sekuntia, jotta laitteen testauksessa ei tarvitsisi odottaa pitkiä aikoja. Todellisessa käytössä kelvolliset ajat olisivat näppäilyaikaa lukuun ottamatta pidemmät, esimerkiksi 30 sekuntia. Näppäilyajaksi kymmenen sekuntia on riittävä.

5 JATKOKEHITYS

Laitteen ja sen ohjelman kehityksen aikana on tullut esiin asioita, jotka tarvitsevat vielä parantelua. Myös ideoita uusista hyödyllisistä ominaisuuksista ja toiminnoista on löytynyt.

5.1 Parannukset

A/D-muunnosten ajoittaminen 50 millisekunnin välein suoritettavaan keskeytykseen aiheuttaa sen, että hyvin lyhytkestoiset sallitun jännitealueen ulkopuoliset jännitetilat voivat jäädä havaitsematta. Toisaalta tämä vähentää riskiä

häiriöpulssien vaikutukseen, mutta se voi myös haitata tärinäkytkimen käyttöä. On kuitenkin epätodennäköistä, että tärinäkytkimen värähtely sattuisi useita kertoja peräjälkeen juuri sille aikavälille, kun kyseistä silmukkaa ei mitata. Parempi ratkaisu olisi kuitenkin suorittaa A/D-muunnoksia lyhyemmällä aikavälillä ja käyttää muunnoksen valmistumisen aiheuttamaa keskeytystä. Näin muunnoksiin käytettävä aika voitaisiin minimoida ja siten saada mahdollisimman lyhyeksi se aika, jona silmukkaa ei valvota. A/D-muunnosten väli on nykyiselläkin ratkaisulla riittävän lyhyt mm. magneettikytkimien tai relelähtöisten liikeilmaisimien käyttöön.

Silmukat voitaisiin varustaa kytkintransistoreilla, joilla ne kytkettäisiin pois käytöstä silloin, kun hälytin ei ole valvontatilassa. Näin poistettaisiin silmukoiden tarpeeton virrankulutus. Tästä olisi hyötyä, mikäli laite olisi varustettu akkuvarmennuksella.

5.2 Lisätoiminnot

Laitteen sujuva ja monipuolisempi käytettävyys edellyttää joitakin lisäominaisuuksia. Näitä ovat esimerkiksi vaihdettava koodi ja muutettavat viiveajat. Näitä varten ohjelmassa tulisi olla jonkinlainen asetusvalikko, jonka kautta kyseisiä asetuksia voitaisiin muuttaa. Valikkoon pääsyn on edellytettävä oikean koodin syöttämistä, jotta asetuksia ei voisi muuttaa luvatta. Valikon voisi käynnistää esimerkiksi jollakin tietyllä näppäimenpainalluksella heti hälyttimen poiskytkennän jälkeen tai jollakin näppäimellä tai näppäinyhdistelmällä, jonka jälkeen oikea koodi olisi syötettävä. Muokattavien toimintojen hallinnan tulisi olla hyvin selkeää ja yksinkertaista, koska laitteessa ei ole näyttöä. Nykyisellä laitekokoonpanolla käyttäjä voi saada palautetta laitteen tilasta visuaalisesti ledien avulla, mutta ne eivät riitä kovin monimutkaisten tilanteiden esittämiseen.

Laitteeseen voitaisiin myös kytkeä pienempi pietsosummeri näppäinääniä varten. Tämä lisäisi käyttömukavuutta, kun näppäimen painaminen aiheuttaisi äänipalautteen. Näppäinäänet voisivat olla poiskytkettävissä asetusvalikon kautta.

Samaa summeria voisi käyttää ledien lisäksi myös äänipalautteen antamiseen helpottamaan laitteen ohjaustoimintoja.

Mikäli laite koteloitaisiin, voitaisiin koteloon kiinnittää niin sanottu kansisuoja eli kytkin, joka valvoo laitteen koteloinnin eheyttä. Sen tarkoituksena olisi ehkäistä laitteen luvatonta käsittelyä, esimerkiksi valvontasilmukoiden eliminointia tai laitteen kytkemistä muuten pois toiminnasta. Kansisuoja voisi olla kytkettynä esimerkiksi sisääntulonastaan RB0, joka voidaan asettaa aiheuttamaan välitön keskeytys, kun sen tila muuttuu. Yhtä lailla kansisuoja voisi olla missä tahansa sisääntulossa, jos sen tilaa tarkkailtaisiin ohjelmassa säännöllisesti ja riittävän usein.

LÄHTEET

1 Microchip. 16F870 datalehti [sähköinen dokumentti]. [Viitattu 20.5.07]. Saatavissa: http://ww1.microchip.com/downloads/en/DeviceDoc/30569b.pdf

2 Fairchild Semiconductor. BC337 datalehti [sähköinen dokumentti]. [Viitattu 20.5.07]. Saatavissa: http://www.fairchildsemi.com/ds/BC%2FBC337.pdf